Specification and evaluation of an assessment engine for educational games

Chaudy, Yaelle; Connolly, Thomas

Published in:
Entertainment Computing

DOI:
10.1016/j.entcom.2019.100294

Published: 31/05/2019

Document Version
Peer reviewed version

Specification and Evaluation of an Assessment Engine for Educational Games: Integrating Learning Analytics and Providing an Assessment Authoring Tool

Yaëlle Chaudy*, Thomas Connolly,
University of the West of Scotland, High St., Paisley PA1 2BE, Scotland, United Kingdom

yaelle.chaudy@uws.ac.uk, thomas.connolly@uws.ac.uk

Abstract

Educational games are highly engaging, motivating and they offer many advantages as a supplementary tool for education. However, the development of educational games is very complex, essentially because of its multidisciplinary aspect. Fully integrating assessment is challenging and the games created are too often distributed as blackboxes; unmodifiable by the teachers and not providing much insight about the gameplays. We propose an assessment engine, EngAGe, to overcome these issues. EngAGe is used by both developers and educators. It is designed to separate game and assessment. Developers use it to easily integrate assessment into educational games and teachers can then modify the assessment and visualise learning analytics via an online interface. This paper focuses on EngAGe's benefits for games developers. It presents a quantitative evaluation carried out with 36 developers (7 experienced and 29 students). Findings were very positive: every feature of the engine was rated useful and EngAGe received a usability score of 64 using the System Usability Scale. A Mann-Whitney U test showed a significant difference in usability ($Z=-3.34$, $p<0.002$) between novice developers (mean=56) and experienced developers (mean=71) but none in terms of usefulness. This paper concludes that developers can use EngAGe effectively to integrate assessment and learning analytics in educational games.

Keywords

Educational games; serious games; games-based learning; assessment; feedback; framework; assessment engine; learning analytics; game development

1. Introduction

Education has evolved in the recent years following a growing access to multimedia technologies. Educators and institutions are increasingly trying and adopting new active ways of teaching such as educational games (Sánchez-Mena, Martí-Parreño, & Aldás-Manzano, 2018). A number of studies about the use of educational games in education show empirical evidence of increased motivation (Annetta, Minogue, Holmes, & Cheng, 2009; Rosas et al., 2003) as well as learning and retention (Gander & Parkway, 2000; Girard, Ecalle, & Magnan, 2013; Squire, Barnett, Grant, & Higginbotham, 2004). Using educational games to assist the learning process offers a wide range of possibilities that can be difficult to attain in a traditional classroom; for example, they give players the possibility of going at their own pace and learning through trial and error in a controlled and safe environment. They also offer many options for assessment; using real-time assessment data, a game can be adapted to a learner's needs (Carvalho et al., 2015; Conlan, Hampson, Peirce, & Kickmeier-Rust, 2009; Göbel, Mehm, Radke, & Steinmetz, 2009). Educational games also offer the possibility of formative assessment and feedback according to a player's actions (Jarvis & de Freitas, 2009). In addition, they can have assessment logic embedded into their core mechanics and offer non-invasive assessment (Kickmeier-Rust, Hockemeyer, Albert, & Augustin, 2008; Shute, 2011; Wiemeyer, Kickmeier-Rust, & Steiner, 2016).

However, integrating assessment and feedback in educational games is time consuming and requires both technical and educational knowledge (Chen & Michael, 2005). For the assessment to be effective and correctly embedded in the game, two different experts are expected to communicate at various stages of the game development. The developer (or development team) is the technical expert in charge of the game mechanics and the educator (or institution) is the subject matter expert responsible for the educational content, the assessment logic and providing appropriate feedback as and when required. Assessment is fundamental in teaching and learning. Learners rely on it to receive feedback on their progress, and

educators need assessment to determine whether their learning goals have been achieved. However, many games are developed with a very basic assessment and no feedback (Chaudy, Connolly, & Hainey, 2013). This could be due to different factors such as a lack of development time, the educators not being fully involved in the development process, the developers focusing on conveying content rather than assessment or the options for assessment integration not being clear to developers and/or educators.

Additionally, the games are too often distributed as "*black-boxes*"; they are closed and self-contained systems, making it arduous to modify or retrieve data from (Serrano-Laguna et al., 2017; Torrente, Del Blanco, Marchiori, Moreno-Ger, & Fernández-Manjón, 2010). This can mean that the potential of the game is reduced since teachers cannot adapt a game to their students' needs or retrieve data about the gameplays to appreciate whether their teaching goals have been met. Educators, developers and researchers have very little insight about what the students learn through an educational game and how they interact with it. Learning analytics (LA) is an emerging field based on data mining processes (Siemens & Gasevic, 2012) that can provide such detailed reports about the use of a game; gameplay data is collected and data mining algorithms and visualisation allow conclusions to be drawn about the game and the players. However, LA is still beyond the reach of most teachers (Johnson et al., 2013).

Various platforms such as eAdventure (Perez-Colado, Perez-Colado, Martínez-Ortiz, Freire-Moran, & Fernández-Manjón, 2017) or e-CLIL (Hainey & Connolly, 2013) provide educators with the ability to create and modify their own computer games; eAdventure even includes a learning analytics module (Martinez-Ortiz & Fernandez-Manjon, 2017). These games engines externalise content and assessment integration from the game's code and partially address the problems identified previously. However, these engines were created for educators alone; they are not meant to be used when working with game developers and therefore only provide limited options in terms of game genres and assessment integration. Teachers sometimes lack the time to develop the games themselves or there is a need for a type of game not offered by such platforms.

In this paper, we propose focusing on assessment at an early stage of an educational game development by using an assessment engine, EngAGe. Our approach is based on the externalisation of the assessment. The engine includes a configuration file where developers describe their game's assessment and a set of web services for performing the assessment. The configuration file guides developers through the assessment integration process by providing guidelines on what should be included in the game, it can also be used as a starting point for communications between educators and developers. The web services then perform the assessment during the gameplay based on the configuration file. Externalising the assessment allows gameplay information to be retrieved and displayed to the educators through a learning analytics dashboard also provided by our system. Developers and/or teachers can then, modify the assessment logic after distribution using an online editor and thus creating different versions of the game.

This paper is divided into six sections as follows. In Section 2, we present a summary of two literature reviews on assessment and feedback integration in educational games. In Section 3, we introduce the assessment framework adopted and the design for EngAGe. Section 4 then explains how EngAGe is used by developers, detailing the configuration file and web services available. In Section 5, we present the evaluation of the engine with developers, finally present conclusions, and discuss the future directions of our research.

2. Previous Research

This section discusses the underpinning research that informed the design of the assessment engine.

2.1 Assessment integration in educational games

This section attempts to identify the different approaches to integrating assessment in educational games and their associated empirical evidence. Search terms include a synonym for educational games: *game based learning* and a broader term: *serious games*. The final search terms used were*: ("serious games" OR "game based learning" OR educational games") AND assessment*. The search was performed on 15 databases relevant to education, information technology and/or social science: ACM (Association for Computing Machinery), ASSIA (Applied Social Sciences Index and Abstracts), BioMed Central, Cambridge Journals Online, ChildData, Index to Theses, Oxford University Press (journals), Science Direct, EBSCO (consisting of Psychology and Behavioural Science, PsycINFO, SocINDEX, Library, Information Science and Technology Abstracts, CINAHL), ERIC (Education Resources Information Center), IngentaConnect, Infotrac (Expanded Academic ASAP), Emerald, Springer and IEEE (Institute of Electrical and Electronics Engineers) Computer Society Digital Library (CSDL).

The literature review covered the period from 2005 to 2015. Relevant papers were identified using the following criteria: papers discussing integrating assessment in an educational game, or papers presenting a framework / engine for assessment in educational games. Papers presenting an educational game without describing how assessment was performed and papers discussing the assessment (evaluation) of the game rather than the player's were excluded. Where possible, the search was based on abstract, titles and keywords to focus on relevant papers. The search on the 15 databases returned close to 3,000, after applying the three pieces of criteria, 34 relevant studies were identified. These studies outlined five main approaches to assessment integration in educational games as described below. This is not a closed categorisation and some of the papers demonstrated more than one approach. The literature search also attempted to identify existing assessment engines. Two papers discussed guidelines and/or frameworks but none proposed an assessment engine.

Quizzes: Some of the educational games described in the papers used an assessment based on integrating quizzes at various stages of the game. The quizzes can be of various types, for example, multiple choice, multiple response, true/false and fill in the blanks. Out of the 34 relevant studies, nine used this approach.

Quests: The assessment of learning can also be part of the game as a particular quest type. Each quest, when completed, can then be assessed. The assessment quests can be, for instance, 'search the Internet', 'create content' or 'match description'. Out of the 34 relevant studies, four used quests for assessment.

Monitoring of States: This approach performs assessment by monitoring meaningful states of a gameplay, the states forming part of an assessment rule. The states monitored can be as general as 'level completed', 'task completed' or 'answer given' but they can also show a more detailed knowledge of the gameplay with 'location visited', 'task started', 'non-player character met' or 'content accessed'. This approach was identified in seven of the 34 relevant studies found.

Use of a Probabilistic Model: This model embeds the assessment within the game and makes it invisible to the player. The probability that the learning outcomes are achieved by a player is calculated based on a probabilistic model. This approach allows assessment to be undertaken without the stress that can be inherent with it, as the student is unaware assessment is taking place. Many of the systems found in the literature search also adapt the game according to the student performances, triggering an intervention from a non-player character (NPC) for instance, or selecting the next mission from a set of possibilities. The models used include Bayesian networks, reverse engineering, petri nets and learning analytics model. This is a popular approach in the literature, with ten studies using it.

Peer Assessment: The last way of assessing knowledge in educational games found in the papers was through peer assessment. This includes informal peer assessment in an in-game chat or forum and players with a special status (e.g. project manager, team leader) providing feedback to other players. This is the least represented approach with only three studies discussing it.

2.1.1 Empirical Evidence

Only six of the 34 studies identified presented empirical evidence of their assessment system. McAlpine, van der Zanden, and Harris (2010) performed a qualitative study to investigate user acceptance of in-game assessment. A game was distributed to schools and played by both teachers and pupils. The testers were then interviewed individually following the trial. Two sets of questions (one for the teachers, one for the pupils) were used as the basis for the semi-structured interviews. Students' answers reflected a clear preference for the assessment in the game against a paper-based one; they considered the process fairer and less stressful. Educators, on the other hand, identified practical barriers to the approach such as requiring a booking of the computer room and software maintenance.

Five of the studies evaluated the accuracy of the assessment system. P. Thomas, Labat, Muratet, and Yessad (2012) carried out a quantitative experiment comparing their diagnostic tool with an online questionnaire. A group of 15 students were asked to play a game on their own for about an hour. After the gameplay, they discovered that they had to answer an online questionnaire developed by their teacher and related to the game's concepts. The students' average for the questionnaire was 1.56 out of 2 and the in-game assessment predicted 1.55. They performed a Wilcoxon signed rank test that returned a p-value of 0.69 indicating that the game's predictions and the questionnaire's answers were quite close.

J. M. Thomas and Young (2010) conducted an experimental evaluation with 16 students of an assessment tool, comparing the results found by *Annie*, their system, with the conclusions of a human observer regarding a prediction of answers to a questionnaire given to the student at the end of the game. On average, *Annie* predicted correctly 76% of the learner answers and the human expert 75% with a student-by-student correlation between both of 0.89 ($p < 0.0001$). They concluded that their in-game assessment behaviour is similar to what could be expected from a human expert.

Csapó, Lörincz, and Molnár (2012) carried out a four-week pilot study with 106 6-8 year olds to provide empirical evidence that an educational game with an Online Diagnostic Assessment System helps compensate for students' learning difficulties. The pupils with performance significantly lower than 50% formed the experimental group (n=42) and the others formed the control group (n=64). The results of the study showed a significant diminution in the difference of means between the two groups for the pre-test (41.7% difference) and post-test (27.6% difference). Over the trial period the control group did not significantly improve ($t = -0.81$, $p = 0.42$) whereas the experimental group did ($t = -9.4$, $p < 0.00$).

Harteveld and Sutherland (2015) conducted an extended study with 145 participants. Their game, Levee Patroller, saves the game data to a remote server as an XML file. The overall game score was computed and the authors analysed how it correlated with other measurements. The results showed that it correlated strongly with both post-knowledge perception ($r = .54$, $p < .001$) and post-test ($r = .71$, $p < .001$) suggesting that (i) the feedback provided in the game may influence the players' perception of their knowledge and (ii) the game score can be used to predict the test performance. The authors concluded that games could be used as an assessment tool, if designed well.

The last empirical study (Lee, Ko, & Kwan, 2013) investigated the use of explicit assessment in games. A game with assessment levels (quizzes) was tested against the same game without. Two studies involving a total of 230 students were performed focusing respectively on engagement and speed. The results were very

conclusive with the experimental group (game with assessment levels) having 30% more completion, being 20% faster and being played for twice as long, therefore engaging more with the game.

2.2 Feedback integration in educational games

Besides identifying how assessment was integrated as a whole in existing educational games, the previous literature review also examined how feedback was supported in games. Most of the studies (29 out of 34) discussed how feedback is delivered through their games. A second literature review was performed to identify papers discussing feedback without assessment in educational games and their associated empirical evidence. The search terms *feedback AND ("serious games" OR "game based learning" OR "educational games")* were entered into the previously listed 15 relevant electronic databases. This second literature review covered the same period and returned 1,549 results. The following criteria was applied: papers presenting an educational game while specifying how feedback was delivered, and papers discussing the different types of feedback related to educational games. The papers were filtered by reviewing title, abstract and full text where needed and 18 studies were identified as relevant. The analysis of these 18 papers combined with the 29 previous ones resulted in a terminology outlined below.

2.2.1 Terminology for Feedback in educational games

Findings from the literature review show that feedback in educational games can be either invasive or non-invasive. Non-invasive feedback triggers an adaptation of the game without the player necessarily noticing it. Invasive feedback can be triggered by a player's action or score. In terms of timing, feedback can either be immediate or delayed. The resulting terminology is outlined in Table 1 and in the following subsection. These characteristics are not closed categories: a single piece of feedback can, for instance, be immediate and provide both guidance and performance information. In terms of media, feedback can be represented as an NPC reaction, videos, sound, haptic feedback (e.g. vibration with mobile phone or Wii remote), images or plain text (e.g. help text, report). In some cases it is even provided by peers.

We focused on the mechanics for triggering and delivering feedback, we did not take into account the content of the feedback message (e.g. process-oriented vs. product-oriented or generic vs. detailed). As discussed in another paper (Chaudy & Connolly, 2018), we believe educators should be able to customise the feedback so any engine should only be able to provide a trigger event for feedback without having to influence its content.

Table 1: Feedback in educational games

Feedback type		Example	Occurrences in the literature
Non-invasive Feedback happens without the player being aware of it.	Game adaptation feedback The player's action triggers a modification in the game	The game difficulty can increase when a player manages to complete the task under a certain time limit, or vice-versa.	13 / 47
Invasive Feedback is obvious and the player notices it.	Guidance feedback Hints are given to help the player.	When a player cannot complete a task, an NPC can provide help and advice.	26 / 47
	Performance feedback	If a player successfully answers a question, he/she can be rewarded with	35 / 47

	Confirmation or correction of the player's answer	extra points, a "well done!" pop-up or fireworks.	
	Learning outcome feedback Triggered when a score reaches a certain limit (low or high)	A game can reward a player with a gold badge when he/she reaches a score of 100%. A special feedback can also be triggered if a score is too low, encouraging the player to practice before trying again.	10 / 47

Immediate or Delayed Feedback

Non-invasive feedback is usually triggered immediately after a player's action or lack thereof. Invasive feedback can either be immediate, triggered 'just-in-time' or delayed, provided at the end of the game or a level, for instance. There are obvious advantages and limitations to both approaches. Immediate feedback can provide information to the players exactly when they need it. Students can, therefore, reflect on their mistakes immediately after having made them, and they can receive further explanation. In their game, Gidget, Lee et al. (2013) integrate quizzes called *assessment levels* that allow the player to reflect on their learning. It is crucial for players to have immediate feedback on the answers selected as they form part of the student learning and contribute to a better understanding and completion of the subsequent levels.

On the other hand, some games will want the player to go through a full sequence of actions, or a whole level, making all the decisions – and possibly mistakes too – before being given feedback. McAlpine et al. (2010) present a game that chose this approach. The players would play the game in full before being presented with a generated report (performance feedback). The students are allowed to play the game as many times as they want until they are satisfied with the report, which they would then submit for formal assessment. In this instance, the aim was for the players to practice and if they had been given immediate feedback they would only require two gameplays, the first one to learn the correct answers from the feedback and the second one to attain a perfect score without achieving deep learning.

Metcalfe, Kornell, and Finn (2009) also argue that, while immediate feedback has its advantages (e.g. avoiding consolidating an error) and many studies favour it, delayed feedback can be better for retention. They suggest that the differences found in previous studies might be due to the learners' attention being more important when the feedback is given immediately after an error compared to delayed. They tested both types of feedback while controlling the *"lag to test"* (delay between feedback and test of knowledge). To force the learners' attention, they asked them to type in the feedback received in both immediate and delayed settings. Two experiments were performed comparing results between immediate feedback, delayed feedback and no feedback on a vocabulary test. The experiment with Grade 6 children showed that results were lowest without feedback and highest with delayed feedback. The same experiment with adults also highlighted the importance of feedback but with no difference between immediate and delayed.

2.2.2 Empirical Evidence

Eleven of the 47 relevant studies presented empirical evidence focusing on feedback in educational games. The resulting conclusions provided a starting point for optimising feedback in educational games.

Feedback should minimise interruptions to the gameplay

Vendlinski, Chung, Binning, and Buschang (2011) compared different media (video vs. graphics) for feedback and instruction given to the players in a maths game. The experiment included a pre-test and a post-test with six groups: a control group, a baseline group with only game mechanics instruction and four

groups with different media for instruction and feedback. The only significant difference ($p < 0.05$) was found in the baseline group, therefore not proving the importance of feedback or a preference in medium. To support the results, the authors cite a warning from Charsky and Ressler (2011) to educators: "*Do not dilute the potential effectiveness of games by taking away the one distinct attribute that gives them their advantage - play*".

Conati and Manske (2009) made similar findings in their study of feedback in a game that teaches factorisation to children. The game initially included a virtual agent helping players with timely interventions. Following a first evaluation, the authors designed and developed a second version of the agent with a more accurate student model. To test both agents, they performed a pre-test/post-test experiment with three groups: without agent, with the old one and with the new one. The scores were analysed using ANOVA and showed no difference in learning between the groups. A questionnaire based on a Likert scale showed that players thought that both agents intervened too often and a time analysis demonstrated that they did not read the guidance feedback.

(Formative) feedback is important

Serge, Priest, Durlach, and Johnson (2013) studied the effects of different feedback content in a shooter game. They compared the evolution of mean score over all four missions of a game and the results of a pre-test and post-test in five groups receiving different feedback: (i) detailed feedback: (ii) general feedback; (iii) adaptive general-detailed (feedback was initially general but switched to detailed if score did not improve); (iv) adaptive detailed-general (feedback was initially detailed but switched to general if the player's score remains high); and (v) control group without formative feedback. The results proved the importance of feedback and showed that all four test groups performed better in the post-test than the control group. The most significant difference was seen with the detailed feedback group and to a certain extent with the adaptive detailed-general feedback group.

Orji, Vassileva, and Mandryk (2013) evaluated feedback qualitatively in an interactive multiplayer game. The game, LunchTime, aims at teaching players how to make food choices from a restaurant menu based on their health goal. The authors gathered evaluation data from surveys and interviews with six participants aged 19 to 40 and concluded that participants found the feedback helpful and the leaderboard motivating. Some participants also explained how the game's feedback made them reflect and even research further as it was sometimes surprising.

Not having access to feedback can help deeper learning

Rick and Weber (2010) investigated competitive guessing games and how players learn to find the equilibrium. Two groups were formed each composed of 38 participants at higher education level. One of the groups received performance feedback and the other one did not receive feedback until the end of the experiment. Results showed that both groups learned, however, the feedback group performed better. In a second experiment, the authors studied the transfer of knowledge to a second competitive guessing game comparing the outcomes of the previous groups to those of a new inexperienced group. Immediate transfer was only observed in the no feedback group and the authors concluded that this approach is more likely to lead to meaningful learning through deeper thinking about the game.

Players can benefit from personalised feedback

In their study, Kickmeier-Rust, Mattheiss, Steiner, and Albert (2011) introduced a variable called 'approach to solution' (ATS) that takes into account if the learner actions are closer to the final solution, further from that or without effect. An experiment was conducted providing 40 learners with four versions of a physics game differing by feedback type: no intervention, adaptive intervention, inappropriate intervention or neutral intervention and compared the ATS values from all four groups. The results showed that the average

ATS (relative to no intervention) had a maximum for adaptive interventions, neutral intervention appeared to be useful albeit five times lower, inappropriate intervention, on the other hand confused the learner rather than help him.

Buschang, Kerr, and Chung (2012) studied different types of feedback in a maths game: a simple performance feedback (the player passed the level) and an individualised guidance feedback (explanation of the player's error). Participants included 187 middle school students. The authors analysed data from three consecutive levels. Although no significant difference was found with the first two, players receiving both types of feedback attempted the third one less times ($p < 0.05$). The authors also concluded that process data can be used to identify the types of errors players make more often and improve feedback.

Rewards can increase learning without decreasing motivation
Filsecker and Hickey (2014) introduced external rewards to their game, badges were added both to their virtual avatar and in the classroom leaderboard. They performed a study with 106 participants from a public elementary school. A survey of players' motivation on a test group with public recognition and a control group without it showed no significant difference in motivation or engagement. The public recognition group, however, showed deeper understanding in one of the three levels tested, the difference between the other two levels were non-significant.

Feedback should make the player reflect on his/her choices
Killingsworth, Clark, and Adams (2015) used an environmental game to compare two feedback types: self-explanation (students were asked to explain their answer) and explanatory feedback (game provides the explanation). They analysed data from 96 students of primary education. Although the self-explanation group performed significantly better on a particular set of questions, both groups showed significant learning gain from playing the game.

Tsai, Tsai, and Lin (2015) tested the efficiency of immediate explanatory feedback (IEF) in a multiple-choice assessment game with secondary school students. They carried out a pre-test/post-test experiment with a test group ($n = 54$) playing a game giving IEF and a control group ($n = 55$) playing without IEF. Results showed that the test group scored significantly higher in the post-test. A questionnaire on participation perception showed no significant difference between the two groups.

Different types of feedback have different strengths and weaknesses
Burgers, Eden, van Engelenburg, and Buningh (2015) looked at the effect of different types of feedback on motivation and play in a brain-training game. Negative (the player did not perform well) and positive (the player performed well) feedback was studied in different forms: descriptive (e.g. "*You train your memory optimally if you complete the game under 60 seconds. You did not achieve this*"), comparative (e.g. "*You completed the game in a time that is above the average of people in your age group*") and evaluative (e.g. "*Poorly done! You completed the game rather slowly, in 87 seconds*"). The experiment was carried out online and included 157 participants not restricted by age, gender or educational level. Their results showed that positive feedback is more effective for long-term play while negative feedback is more effective for short-term play, motivating the player to try again immediately. Future gameplay was increased with evaluated feedback and decreased with comparative feedback.

3. The proposed approach: An Engine for Assessment in Games

As discussed in Section 1, there are three key problems to integrating assessment in educational games: (i) assessment in educational games requires significant time, effort and communication between both educational and technical experts, (ii) educational games are often distributed as "black-boxes"; the

assessment is embedded into the game's code and educators cannot modify it to suit their students' needs, and (iii) there is a lack of detailed reports about the students' interactions during the gameplays and their assessment. In this section, we present our approach to addressing these three problems. A framework was designed for integrating assessment, feedback and learning analytics in educational games. The framework was then used to create an engine, EngAGe (an Engine for Assessment in Games), to be used by both educational games developers and educators. The proposed approach addresses the first problem by separating the assessment from the game. It creates a link between teachers (editors of the assessment) and developers (implementers of the assessment) with a configuration file describing the game's assessment. This file is based on educational concepts, understandable by a teacher and easy to write for a developer. Web services, called from the game, then perform the assessment and return relevant information to the game.

With the assessment being external to the game, it is now possible for educators to modify and adapt it to their students' needs, even after the end of the development process and without having to edit the source code. This functionality addresses the second problem. It is achieved through an online visual assessment editor. Teachers can also visualise their list of games along with their unique versions, manage their students' access and share their games with other teachers. Having this control over the games will help educators develop a sense of ownership and trust towards the tool. To address the last problem, the assessment engine includes a learning analytics (LA) dashboard; all the data processed by the engine will be stored allowing for data mining on all the gameplays of a particular game and across games. The aim is to make LA more accessible to teachers, developers and researchers. Educators can then make informed decisions about the changes needed to adapt their games.

3.1 A framework for integrating assessment, feedback and LA in educational games

The proposed framework is based on the literature reviews previously presented and on the preliminary model proposed by Hainey, Connolly, Baxter, Boyle, and Beeby (2012) for assessment integration in educational games. The model is based on the Input/Process/Output Game Model (Garris, Ahlers, & Driskell, 2002) and focuses on the assessment aspect. Figure 1 shows the proposed framework, which comprises of four phases: Planning, Design, Gameplay and Revisions. All the components of the planning section, with the exception of the institutional content, should be made customisable by the teacher during the revision phase.

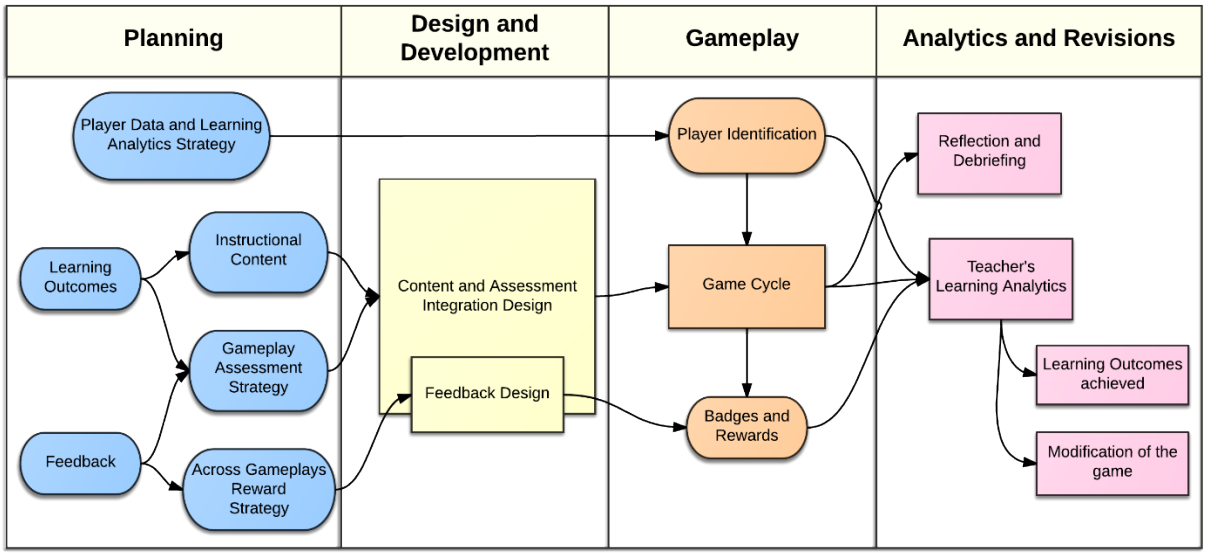


Figure 1: Framework for integrating assessment and LA in educational games

3.1.1 Planning

The planning phase has the following steps:

- Specification of the *learning outcomes:* what the player should learn from the game.
- Specification of the *instructional content* corresponding to the learning outcomes.
- Specification of the *feedback* of the game: what reactions are possible based on the players' actions? Feedback can have various forms such as messages, adaptations and badges; they can also trigger the end of the game.
- Specification of the *gameplay assessment strategy*: should the overall strategy be to facilitate formative assessment, summative assessment or a combination of both? What information, from the gameplay, should be used for assessment (e.g. states, actions, answers to quizzes)? What actions from the player should trigger feedback, and should it be immediate or delayed?
- Specification of the *across-gameplays reward strategy*: when planning a game, feedback across gameplays should also be considered. What states should trigger that feedback?
- Specification of the *player data*: what is relevant information about the player (e.g. age, language, gender)? This data will be available to the teacher when generating learning analytics.

3.1.2 Design and development

The design phase involves integrating the pedagogical content with the assessment, which involves both educational experts and technical experts. During this phase, the integration of the assessment is implemented along with how this should be displayed (e.g. text, graphics and animation). At this stage, the type(s) of assessment should be defined. Feedback should also be considered and this phase will address how the feedback is shown to the player for both immediate (e.g. colour, pop-up, NPC talking) and delayed feedback (e.g. report, log). Badges or other rewards should be designed and should have a place in the game (e.g. in the menu screen before starting the game or as new unlocked levels).

3.1.3 Gameplay

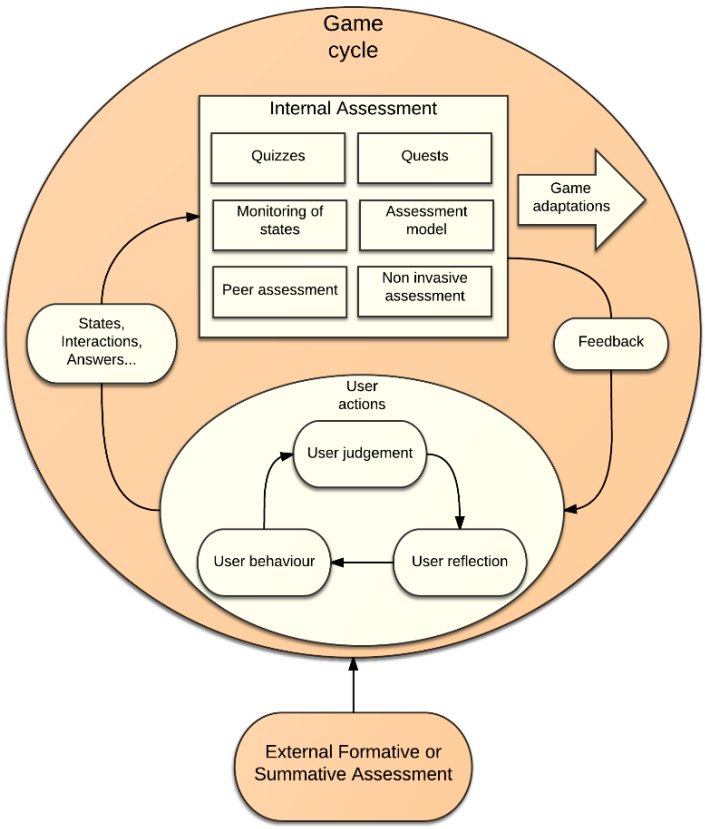


Figure 2: Detailed Game Cycle Model

The gameplay phase corresponds to what happens after distribution of the game. Before starting a game, the student logs in and provides relevant information specified in the player data step of the planning phase. Once logged in, the gameplay can start. The game cycle step is described in more detail in Figure 2. During the course of the gameplay the students will use their judgement, reflect and behave in reaction to the content and the rules of the game. Information about the student's interactions with the game will be sent to the internal assessment module. External assessment can be performed outside of the game cycle, either periodically or at the end. The assessment of a student's interactions can trigger game adaptations or more invasive feedback. After a gameplay, the player's overall performance is revised across all play sessions. Badges can be attributed or special levels can be unlocked based on various criteria, such as the number of times a student played, the total time spent playing, the number of gameplays won or the game scores.

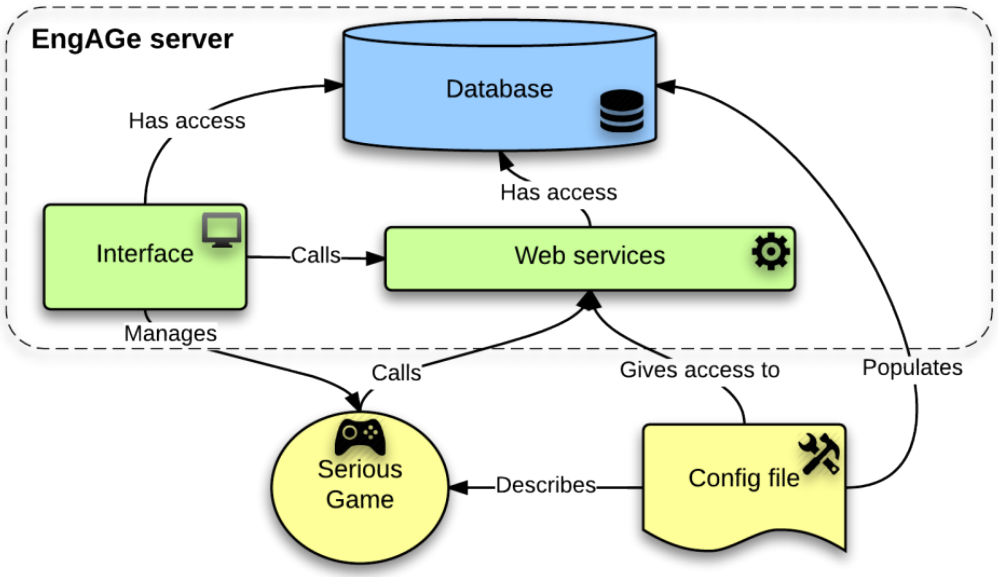
3.1.4 Analytics and Revisions

When the game is complete, students can reflect during a debriefing period with peers and instructors. External summative assessment could optionally take place at other times, for example, at the end of a learning unit. Teachers have access to the gameplay data through learning analytics (LA). They can use this to confirm that the learning objectives have been achieved by their classes. If this is not the case, the LA also provides them with relevant information to identify either issues with the game assessment (e.g. tasks that are not adapted to the level of the players, questions that are ambiguous, quests that are not challenging enough) or with some players that might be at risk. After reviewing the LA, teachers can decide to modify the game assessment to fit their students' needs, this should be done via a visual editor.

3.2 EngAGe's architecture

EngAGe was designed to apply this framework easily to a wide variety of games. Its architecture, presented in Figure 3, includes the following:

- A domain-specific language (DSL) to be used when writing the game's configuration file.
- A set of web services to parse the configuration file and perform the assessment.
- An interface to visualise the data collected during gameplays, manage educational games and modify the assessment.



3.1 Figure 3: Architecture of EngAGe

3.1.1 Games supported by EngAGe

At the time of writing, the engine supports games applying 4 out of the 5 types of assessment identified in section 2.1: Quizzes, Quests, Monitoring of states, and Use of probabilistic model. Peer assessment is not yet fully supported as the system is only able to update one player's score and send him/her feedback: for EngAGe's integration to be successful the game must be synchronous and update a player's score and feedback from the side of the player receiving the assessment. In terms of games genre, most are supported with the exception of asynchronous multiplayer, EngAGe however is limited to games that have an internet connection (e.g. online games or mobile games on devices connected to the internet) in order to perform the web services calls.

4. Integrating EngAGe in educational games

EngAGe is designed to be used by developers when creating an educational game. Using EngAGe is a two steps process for developers, as discussed in this section. Section 4.1 explains how a configuration file can define a game's assessment and Section 4.2 details EngAGe's database schema and the set of web services available for performing the assessment during the gameplay.

4.1 Describing the game's assessment in a configuration file

4.1.1 Using a domain-specific language

Configuration files, in general, can either use an existing format such as XML (Extensible Markup Language) and JSON (JavaScript Object Notation) or follow a pre-defined grammar with a domain-specific language (DSL). Fowler (2010, p. 27) defines a DSL as: "*A computer programming language of limited expressiveness focused on a particular domain*". For EngAGe, a DSL is an appropriate approach for two key reasons. Firstly, it allows the domain (assessment in educational games) to be embedded into a programming language; knowledge of assessment integration is situated in the engine itself and no longer in the game. Guidelines can therefore be provided on assessment in games by the DSL structure. Secondly, the assessment logic (defined in the DSL) can easily be separated from the game mechanics (programmed in any general-purpose language) as long as an interface is provided for them to communicate. This will allow for the modularity needed for educators to modify the assessment without access to the game's code. EngAGe therefore relies on a DSL for the configuration of assessment. It is an external DSL with a simple compiler that will convert the configuration file into a JSON object for easy manipulation and communication with the game.

4.1.2 Model for the EngAGe DSL

EngAGe's DSL model comprises seven sections presented in the syntax diagram in Figure 4. The DSL itself should be seen as a guideline to the assessment process with the intention to encourage educational games developers and educators to include details of the assessment in their games they might not think about such as feedback, adaptations or badges. However, in order to make the DSL more usable and generalizable, a number of the components of the language were made optional as they are currently omitted in numerous existing educational games (Chaudy et al., 2013).

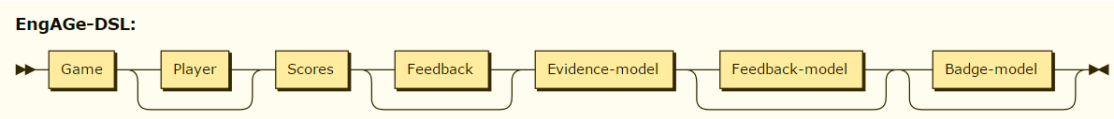


Figure 4: Syntax diagram of the DSL

Game description: The first section of the configuration file describes the educational game; it takes the form of a list of characteristics. The section itself is mandatory but there are only two compulsory parameters: a name, for identification purposes, and a developer id, for access to the game management interface. Other optional parameters include: description of the game, target player's age range, language, country, genre and subject matter taught. The game designer can also specify whether their game is public (i.e. can be played by unidentified guests). The optional parameters are used by the system to facilitate search and indexation of games.

Player characteristics: For learning analytics purposes the system needs to have knowledge of certain characteristics of the player such as his/her gender and age. As every educational game is different, there is no exhaustive list of meaningful player characteristics. The player section therefore allows definition of any relevant player characteristics. A characteristic has a name, a type and the question to ask the player.

Scores: This section allows the definition of the game's learning outcomes and other scores not necessarily associated with instructional content and learning (e.g. the number of lives a player has, the quantity of fuel). Every score has, at a minimum, a name and optionally a description and a starting value (float, default value is zero) can also be specified. After defining a score in this component, its name will be available for use as a reference in the subsequent blocks.

Feedback: Feedback is very important in the teaching and learning process and it is logical to see it appear in the language as a component. Every piece of feedback is given a name and a message and optionally can have a type (positive, negative, badge, hint and adaptation). Feedback can also trigger the end of the game by specifying the keyword win (game won), lose (game lost) or end (end of game). A piece of feedback is only defined once here and its name is used as a reference in the subsequent blocks.

Evidence model: The terminology 'evidence model' was introduced by Zapata-Rivera, Hansen, Shute, Underwood, and Bauer (2007) to refer to the logic of the assessment and how what is observed provides evidence about the learning. This component defines the meaningful actions of the game from an assessment point of view and describes how the scores are updated after them and if feedback is triggered as a result. Every action contains a name, parameters, a set of consequences and a reaction block. A consequence is composed of a list of values for the parameters of the action and the rule to update the score if an item of this list is chosen. The reaction block allows the game developer to specify rules for triggering feedback.

In-game feedback model: In this block, the general feedback model within a gameplay is defined, however, this time the trigger of the feedback is not an action but a broader concept such as the reaching of an upper or lower limit for a score or a significant inactivity perhaps indicating that the player is confused. The developer defines the meaningful events for the game and associates the feedback to be sent.

Across-gameplays badge model: While some badges are earned during the gameplay some others need more global information, such as the number of times a player has played the game or the total time spent playing. This across-gameplay feedback is defined in this block. There are two types of trigger for badges: basic ones, based on the number of gameplays and time, and the ones based on scores.

4.1.3 A mini game to illustrate the DSL

In this section, we illustrate EngAGe's DSL using a simple mini game. EU mouse was implemented to demonstrate the potential of the engine and to develop a tutorial for developers. It is an endless runner type game where the player is a mouse running through a geography classroom. The mouse has to collect the countries that form the European Union (EU) scattered in the room. The player is given three lives and loses one when collecting a country that is not part of the EU. The game keeps track of the countries found and the player wins when he/she collects all 28 correct ones. Figure 5 shows a screenshot of the gameplay and Figure 6 represents the assessment configuration file of the game. For brevity, the evidence model has been intentionally shortened, with the EU country list not shown in full.

Figure 5: EU mouse gameplay

```
Game
   name : "EU mouse"
   developer : 1
   desc : "This is a mini serious game that teaches you to identify countries part of the European Union"
   ageRange : 10-99
   language : EN
   country : UK
   genre : "runner"
   subject : "geography"
   public : true
End
Player
  age Int "How old are you?"
  grade String "What grade are you in?"
  gender Char "Are you a boy or a girl?"
End
Scores
  eu_score "overall score, number of correct countries identified" 0
  eu_countries "distinct countries of the EU left to find" 28
  lives "number of lives the player has" 3
End
Feedback
  correct_country "Well done, [country] is indeed part of the EU" positive
  wrong_country "Nope, [country] is not part of the EU" negative

  gold_medal "You found 200 EU countries" badge
  dedication "You played more than 60 minutes" badge
  effort "You played 5+ times" badge
  performance "You won 10+ times" badge

  speedGame "You're good, let's make it more challenging" adaptation
  slowGame "Seems you are strugling, let's slow things down" adaptation

  end_win "Well done, you won the game :)" win
  end_lose "Sorry, you lost the game :(" lose
End
Evidence-model
  newCountrySelected( String country )
  "When a player selects a country for the first time"
    eu_countries -> -1, eu_score -> 1
        austria
        [...]
        united_kingdom
    End
    lives -> -1
        others
    End
    When
        any (+) : correct_country immediate
    End
  End
  countryReSelected( String country )
  "When a player selects a country he/she had already selected"
    eu_score -> 1
        austria
        [...]
        united_kingdom
    End
    lives -> -1
        others
    End
    When
        any (+) : correct_country
        any (-) : wrong_country
    End
  End
End
Feedback-model
  eu_countries < 1 : end_win
  lives < 1 : end_lose

  eu_score > 60 : speedGame
  lives < 2 AND eu_countries > 20 : slowGame
End
Badge-model
    numberGameplays > 4 : effort
    numberWin > 9 : performance
    totalTime > 60 : dedication
    sumScore eu_score > 199 : gold_medal
End
```

Figure 6: Configuration file of the mini game EU mouse

4.1.4 An editor for the DSL

To assist developers in writing their configuration files, an editor was created. An important characteristic of EngAGe is that it does not require any new software, downloads or installation. In order to maintain that characteristic, the editor was developed as an online tool and integrated into the EngAGe web interface. The editor includes a syntax highlighter and a live syntax checker showing, in real time, possible parser errors. Figure 7 shows the editor in use within the EngAGe interface. In the editor page, developers can modify their configuration file, save it and re-load it later.

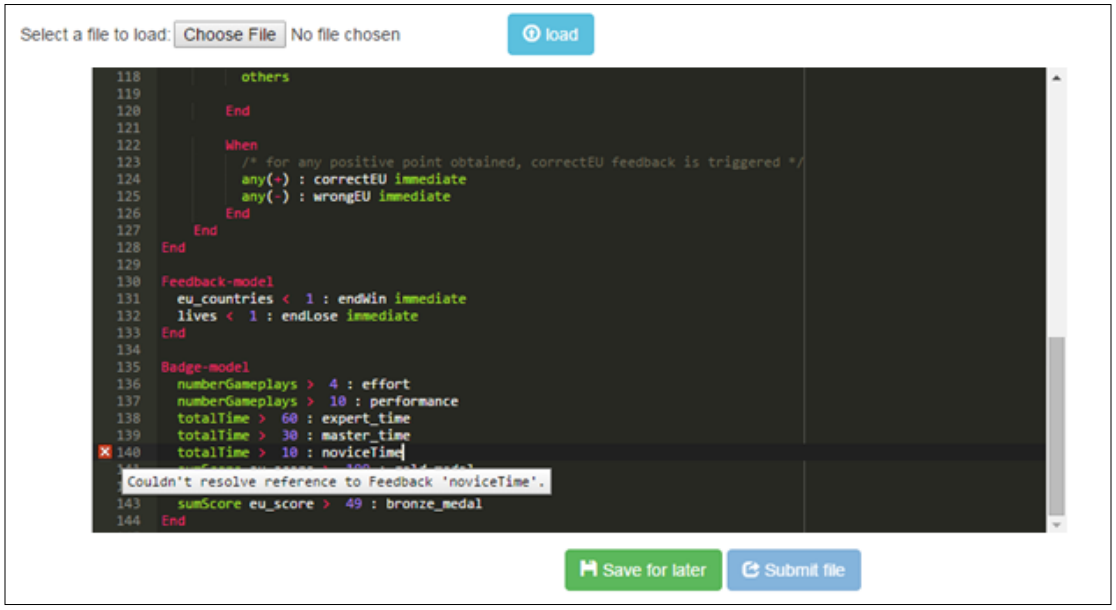


Figure 7: EngAGe online editor with its syntax checker

4.2 Performing the game's assessment with a set of web services

After defining a game's assessment in the configuration file, the developer can submit it via the editor in the interface. This action will populate a database with the new game data, as described in Section 4.2.2. Once the game is instantiated, a set of web services are made available for performing the assessment and retrieving important information. These web services are described in Section 4.2.3.

4.2.1 A service-oriented architecture

In terms of architecture, the two main options for implementation of the engine were a library and web services. The latter option was elected for three main reasons. Firstly, web services offer more flexibility in terms of the programming language used for the game development; there is no need to restrict the language as a web service can be called from many types of applications. Secondly, they offer the possibility of creating a common repository in which all the gameplay sessions can be recorded. This enables data mining for detection of issues with the game and with the learning process amongst players globally, potentially for all games that use the assessment engine. Finally, because EngAGe is still evolving, web services were the best option as they allow for new versions of the project to be deployed without the educational games developer having to upgrade the tool on his/her environment. Figure 8 presents an overview of the architecture featuring target users, services and resources.

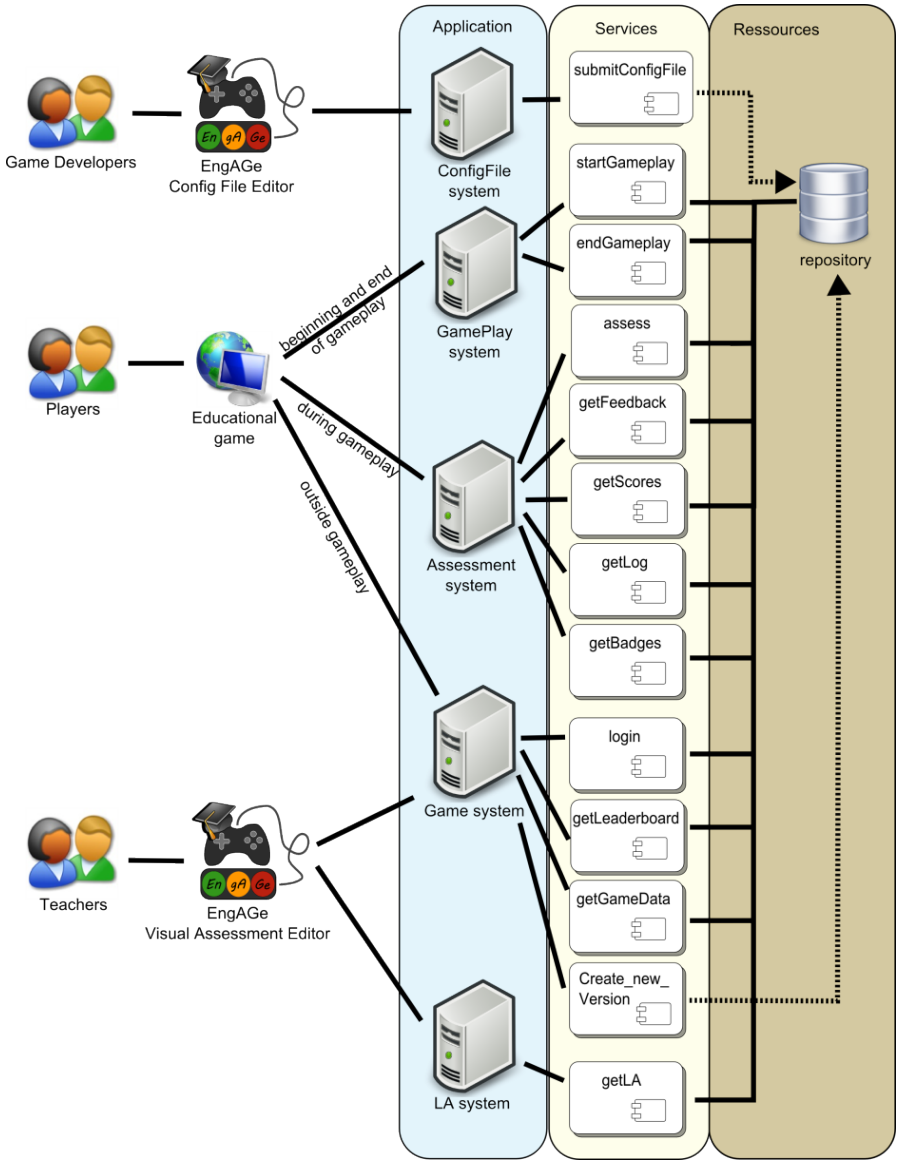


Figure 8: Overview Architecture of EngAGe

4.2.2 Parsing the DSL and populating a database: Creation of a game

At the creation of the game, the developer defines the configuration file and sends it to the engine by submiting it through the online editor. The *submitConfigFile* web service first saves the original configuration file into a *config_files* table in the database. This will give developers access to all their previously submitted configuration files through the interface. Then the function saves the game details in a *game* table. The table is linked with a many-to-many association to another table, *developer*. When a game is created in the database, it is associated to a developer via a *game_developer* table.

There is no existing player table as every set of characteristics required is unique. Instead, the player section of the configuration file is used to generate a new table for the description of a player. The table is called *player_[idGame]_0* where *[idGame]* is the id of the game it belongs to and 0 represents the first version of this game. The teacher or developer will have the ability to modify a game, thus creating new versions. The scores section of the configuration file populates a *score* table and the feedback section creates rows in a

feedback table. The three last sections of the configuration file, corresponding to the evidence, the feedback and the badge models are too complex to be represented in the database, instead the web services use the raw JSON.

The *submitConfigFile* web service, if successful, will return the game id and will make a set of web services available. These web services are capable of processing a specific communication protocol to perform the game's assessment. Once the id is received, the game can be implemented using the web services and communication protocol.

4.2.3 The web services available to the game developer

There are 12 main web services available to a game developer, some can be used before the gameplay, some are called during the gameplay to update the assessment data, and some others are useful after a gameplay, as a debriefing phase. All the web services are described below.

Before the gameplay: A teacher or developer can modify the game's assessment, thus creating a new version of the game. Before the player can start playing, the game needs to retrieve the version the teacher decided they would play using the login web service. If the student has never played before, the service will also return the list of questions defined in the player section of the configuration file. As the game data can also be modified by the teacher, a *getGameData* web service is available to the game developer that returns a JSON object with the game's details. This information may be used in the game to update an "about" window or to set up a welcome screen.

The gameplay: During the gameplay, the game will communicate with the engine to perform the assessment and retrieve feedback. Six web services are available to developers. The first request that needs to be sent is to the *startGameplay* web service. This web service needs to know the id of the game, its version and the player's information. The service creates a new gameplay session and returns its id. During the game, the player's actions are assessed by invoking the assess web service. This web service receives the id of the gameplay and the action performed by the user. Using this information and the evidence-model section of the configuration file, the service will update the player's scores, log the action and possible feedback in the database and will return a JSON object containing the updated scores and feedback that the game can display.

At any time during the gameplay, a game can ask the engine for any feedback it should provide to the user; this may be outcome or inactivity feedback that can be retrieved separately rather than through an action that has triggered automatic feedback. This is done by invoking the *getFeedback* web service with the id of the gameplay. The game can also call *getScores* to receive the last version of the gameplay scores. The developer might also need access to the configuration file and especially the list of possible parameters for a specific action, which can be retrieved using a call to the *getParamsAction* web service with the name of the action. Finally, when the gameplay ends, the game needs to notify EngAGe with a call to the *endGameplay* web service. This will update the database, adding the end timestamp and setting the win Boolean in a *gameplay* table.

After the gameplay: As part of a debriefing after a gameplay, there are various options for game developers. The game can display a personal summative assessment, such as a report of the player's actions stating how well he/she performed, what mistakes were made and the final scores. This information can be retrieved from EngAGe with a call to *getLog* with the gameplay id. It should be noted that this service can also be used during the gameplay to adapt the game based on the player's actions. The developer can decide to retrieve more succinct information: the list of badges earned by a player is returned by *getBadges*, the game's leaderboard by *getLeaderboard*. Finally, for more detailed information about the game's use, *getLearningAnalytics* can be called with a game id and a version. This web service returns extensive

information about the players and the gameplays. This data can be used to identify flaws in the game or students at risk. EngAGe's learning analytics dashboard provides a visual representation of this data.

4.2.4 Data protection and security

Security and data protection were taken into account when building the engine. Web requests are sent over HTTPS guaranteeing that data is encrypted and secure and EngAGe is currently deployed on Azure. Since May 2018, the General Data Protection Regulation (GDPR) is in effect in the European Economic Area (EEA). The GDPR defines two roles in data protection: data controllers and data processors. In the case of an educational game developed with EngAGe, the developer together with educators would act as data controller; they control the data through the configuration file and assessment editor, they define what is stored and they are in charge of interactions with the user. EngAGe acts as data processor; it stores the data sent by the game and process it in the LA dashboard. Hoel, Griffiths, and Chen (2017) summarise the requirements related to LA processes following GDPR. The requirements relevant to data processors include:

- **Right to access**: players can access their personal data
  - EngAGe provides a player view of the LA dashboard and the getLog() web services
  - Developers should provide a way for the players to access them
- **Right to data portability:** for the players to access their learning activity data
  - EngAGe provides the getLogs() web service
  - Developers should provide an interface for the user
- **Rights to object**: players can opt-out of the process
  - EngAGe provides a functionality for "guest" gameplays. All game sessions logged by guest players are recorded without any personal data; if a guest player returns it will count as two different players. While that might be a solution for some games, if the LA are required, (e.g. for assessment of a course), they would not provide sufficient data however.
  - Developers should provide an "opt-in" button for the player to agree with the terms of service. When used solely in a course, the opt-in option could also be included when the player enrols in a course.
- **Right to be forgotten**: players can ask for their information to be deleted and no longer logged
  - EngAGe gives the option to anonymise the player's game session through "guest" login for future sessions.
  - Educators and developers should decide whether EngAGe functionalities are required for a course and either anonymise the player's data or propose that the player terminates the course.

At the time of writing, EngAGe has only been used when closely working with game developers which made GDPR compliance. Future work will include a set of tutorials to explain to developers and educators their responsibilities and what EngAGe proposes on its part.

5. Evaluation of the engine

EngAGe was designed to be used by two target users: developers creating the games and educators using and modifying them. The authors evaluated EngAGe quantitatively with 31 educators (Chaudy & Connolly, 2018). The study shows encouraging results with both the assessment editor and the LA dashboard rated useful and easy to use. The evaluation also compared educators' trust towards educational games as assessment tools and found that, having access to EngAGe, educators would be more likely to trust a game's assessment.

This paper will cover the evaluation with developers and will attempt to confirm that (i) the engine can be used with a wide variety of games; (ii) developers find it useful; and (iii) it is relatively easy to use. Section 5.1 presents the evaluation of EngAGe's generalisability with a case study and Section 5.2 evaluates quantitatively the usability and usefulness of the engine with 36 game developers.

5.1 Generalisability

One of the key benefits of EngAGe is its generalisability. It is an important requirement that the engine can be used in a variety of educational games. The main criterion for a successful integration of EngAGe in any game is the description of the game's assessment using the DSL created for the configuration file. This section presents the results of a pilot study performed in Autumn 2014.

5.1.1 Methodology

Participants

The participants were games technology Honours students enrolled in a Serious Games module at Higher Education level. Each student had a minimum of three years programming experience in C++/Java. The module requirements include the specification and development of a serious game. The students are free to create any game providing its primary objective is not entertainment. The class was composed of 13 groups of students (a group typically contained 2-4 students).

Design of the study

During the first weeks of the module, the students were asked to decide on an educational game to develop and pitch it to the teaching team, who act as games producers. EngAGe was only presented to the students after the pitches to make sure that the engine did not influence their decision on the games and the assessment logic. Several tutorial sessions were organised to explain the purpose of the engine and describe the configuration file structure. Students were then asked to use the DSL to describe their game's assessment in a configuration file. They were also given the option to use the engine within their games. Guidance was provided throughout the semester. The methodology used included regular observations as the configuration files were being created and analysis of the configuration files produced.

5.1.2 Results on the generalisability

All of the 13 groups in the second study defined their game's assessment using the DSL, 12 submitting a final working version. The DSL editor includes a '*save*' and a '*submit*' button. Both buttons save the configuration files created to EngAGe's database. During the time of the study, the engine recorded 121 configuration files (52 submitted and 69 saved). We compared the different versions of the files for each group and drew conclusions on the file creation process, the difficulties with the DSL and the errors most commonly occurring. Table 2 presents the 13 games created by the participants. General conclusions about the DSL drawn from the comparison of configuration files and observation of groups include:

- The editor helped developers fix the file's syntax errors on their own.
- The DSL was able to describe all of the proposed games' assessments.
- The first two sections of the DSL were easy to write and helped developers get started with defining their configuration file.
- The DSL structure steered discussions about educational aspects of the games (early group discussions changed focus from game mechanics to education). Some assessment elements such as badges and feedback were added to the original game design.
- Many participants did not feel confident submitting the file on their own and waited for approval first.

- The configuration files created were sometimes lengthy, listing all possible game choices as an action in the evidence model with a Boolean 'correct' as a parameter. Although such a file would not prevent the engine being used, it does not delegate the assessment to the engine. It was concluded that EngAGe's documentation should include a section on best practices with examples of common assessment types.
- Participants with little experience of programming found the DSL difficult to use whereas the more experienced programmers succeeded in creating a correct version quicker and without help.

Table 2: Games created by the participants in the second study

Game	Assessment	Config files saved - submitted
IT management	Based on quests, the game teaches project management skills within the IT industry. It has two learning outcomes: efficiently managing the staff and efficiently meeting targets. The player is the project manager, assigned a number of projects to manage, and he/she has to hire the best team and select the best employee for each task. He/she also has to resolve conflict in the team. The game includes performance feedback, badges for each project completed and guidance feedback.	4 - 7
Seriously Vector	Based on quizzes, the game has two learning outcomes: adding vectors and subtracting vectors The player is asked to answer questions about vectors. A good answer allows him/her to progress to the next part of the game world. Performance feedback is triggered to confirm/inform the player's answer.	6 - 3
Interview Sim	Based on multiple choice quizzes, the player is taken through a simulated interview and has to answer examiner's questions. The game monitors how well the questions are answered and provides feedback on them.	11 - 2
Space Physics	Based on multiple choice quizzes, the player is asked to answer questions about space physics. The game includes various learning outcomes about understanding and applying the principles of Newton's Third Law, Projectile Motion and Inertia. The player's actions trigger performance feedback and badges.	8 - 11
Disease Awareness	Based on a monitoring of states and quests, the game teaches disease awareness. The player has to save the world from a pandemic disease, first by identifying the disease then by taking actions to stop it. The game triggers performance and guidance feedback.	3 - 1

Be your own boss	Based on a monitoring of states model, the game is a simulation of the creation of a game company. The game's only score is the money the player has. According to the time of the year and the player's choices the game triggers earnings (from the games created) and spending (rent, salary, etc). No feedback or badges are included.	6 - 2
Episodic Memory Test	Based on quests, the player is taken twice through different rooms. The second time he/she is asked to remember where certain objects were initially placed. The game was developed to compare the results of players playing on different displays (desktop monitor and head mounted display).	1 - 9
Geo Explorer	Based on multiple choice questions, the game tests the knowledge of geographic features. Performance feedback and badges based on progress are triggered.	4 - 5
Deep space typing disaster	Based on quests, the player has to defend the planet against asteroids by quickly typing words to destroy them. The players are assessed based on how many words per minute they can type.	3 - 1
Doctor Sim	Based on quests, the player plays the role of a doctor and is presented with sick patients. He has to form a diagnostic and provide the correct medicine for each one. The game has one score corresponding to the number of patients cured. It triggers performance feedback and badges based on outcome score.	10 - 7
The French Mystery	Based on quests and multiple-choice questions, the player is a detective tasked with solving a mystery by interrogating suspects. He has to prove his/her knowledge of French in doing so. The game has one score for the French knowledge, every correctly answered question increases this score. No feedback or badges are included.	1 - 3
The life skills game	Based on monitoring of states, the game has four scores: money, budgeting skills, communication skills, and happiness. The player has to make decisions on a life simulation based on the money he/she has. The decisions impact on the different scores. The game can trigger feedback based on the player's score congratulating him/her on particular skills or explaining how to improve.	6 - 1
Mental health	Based on quests, this game tests the player's mental health with different tasks such as ordering the alphabet and answering riddles and puzzles.	6 - 0

All the configuration files submitted described correctly the games' assessments. Particular attention was given to the '*Mental health'* game as it is the only game for which no configuration file was submitted. However, the last configuration file saved for this game was syntactically and semantically correct and could have been submitted even though it only covered one of the game's quests. We were able to write the missing sections corresponding to the remaining quests, therefore proving that the DSL can be generalised to this game.

In conclusion, the study showed that, even if the participants needed help creating their configuration file, it was possible to define the games' assessment using the DSL. The weekly observations also suggested that the configuration file's structure made participants include assessment elements (badges and feedback) that were not in the original design. The study also served as a formative evaluation; comments from the participants were used to improve the DSL's grammar, EngAGe's interface and its web services.

5.2 Usability and Usefulness

The next step towards a formal evaluation of the engine was to carry out a study to ascertain whether the engine can successfully be integrated by educational games developers into new games and by a variety of developers. A summative evaluation was carried out to determine whether integrating EngAGe into educational games was easy for developers and how useful they found the process.

5.2.1 Methodology

An ideal evaluation study would have educational games developers create any educational game in any programming language while integrating EngAGe. However, this was not possible in the scope of this research; finding a large number of educational games developers willing to spend time and effort creating a game while using a new tool was too arduous. Instead, a tutorial was created to teach developers how to create an assessment configuration file and integrate EngAGe's web services into the existing EU mouse mini-game. The code for the game without assessment was provided. After completing the tutorial, the developers were asked to complete a questionnaire specifying: 1) the time spent doing the tutorial; 2) their opinions about the usefulness of the configuration file and the web services available; 3) their opinions regarding the usability of the tool using the SUS questionnaire.

Participants

In order to gather sufficient answers for a quantitative analysis, participants included three main groups. First, the tutorial and associated questionnaire were distributed online via email to experienced developers (n = 7). Then it was presented to games technology Honours year students (n = 15) enrolled in a Serious Games module who had a minimum of three years' games programming experience and some experience with educational games. Finally, it was introduced in a face-to-face lab in a Games Programming module (n = 14) where the students had only a limited experience with games development and programming in general. A flaw of online questionnaires is the lack of validity of the answers (Wiersma, 2011). To avoid unreliable answers, the questionnaire includes a field corresponding to the developer ID of the participant, this allowed us to verify that 1) participants only answered once and 2) participants actually completed the tutorial. EngAGe's interface includes a monitoring tool for administrators allowing them to visualise the configuration files submitted and the number of gameplays. A total of 36 valid questionnaire returns were recorded.

5.2.2 Usefulness

After using the engine, participants were asked to reflect on the usefulness of the tool for designing and performing a game's assessment. They rated EngAGe's features on a five-point Likert scale between *Extremely Useful* and *Absolutely Not Useful*.

The configuration file

The evaluation questionnaire includes a section asking participants to individually rate the usefulness of the seven sections of the DSL on five-point Likert scales. The question introducing the section was: "*How useful did you find the following sections of the DSL for reflecting on and designing the game's assessment?*". Figure 10 displays the answers of the 36 participants in a diverging stacked bar charts as recommended by Robbins and Heiberger (2011).

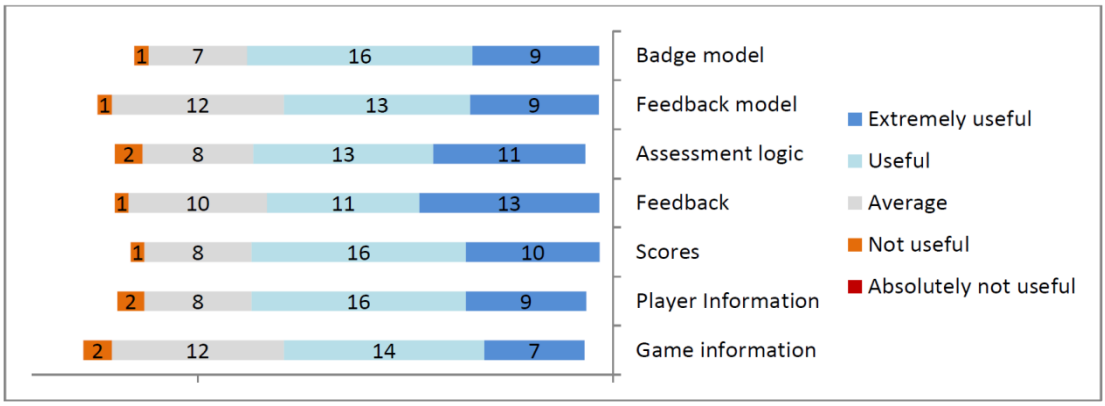


Figure 9: Usefulness of EngAGe's DSL sections

The participants' ratings on the DSL usefulness were scored from 5 for *Extremely Useful* to 1 for A*bsolutely Not Useful*. The ratings were overall very positive with only three participants grading some sections not useful. The mean ratings show that all sections are considered above average. Table 3 presents the results ranked from most useful to least useful with their mean rating and standard deviation (SD).

Table 3: Usefulness - Ratings of EngAGe's DSL sections

Section	Rank	Mean	SD
Feedback	1st	3.92	0.87
Scores	2nd	3.89	0.8
Player Information	3rd	3.81	0.85
Assessment logic	4th	3.75	0.9
Feedback model	4th	3.75	0.83
Badge model	6th	3.67	0.79
Game information	7th	3.64	0.85

The web services: Performing the assessment

The eight main features of EngAGe, all associated with a web service, were evaluated in terms of usefulness in the questionnaire using a five-point Likert scale. Figure 11 presents the participants' answers.

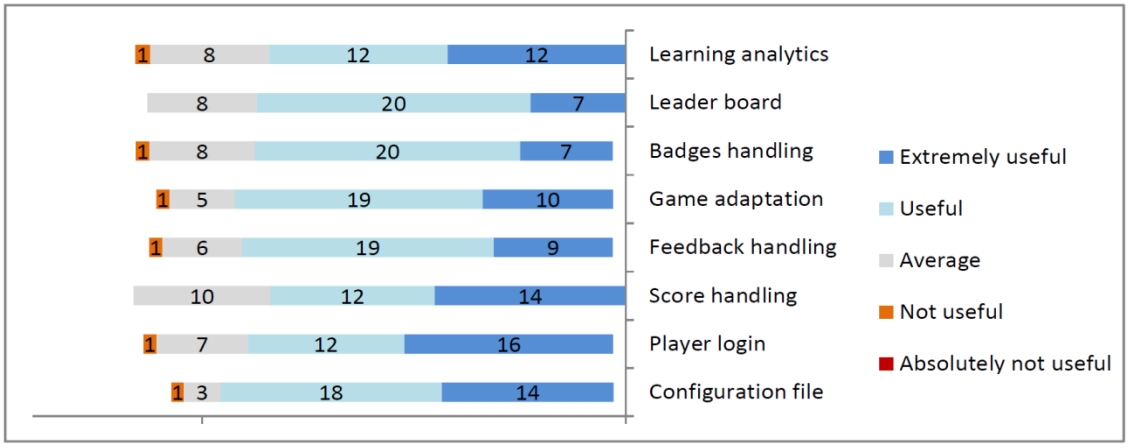


Figure 10: Usefulness of EngAGe's web services

The participants' ratings of the web services usefulness were scored from 5 for *Extremely Useful* to 1 for A*bsolutely Not Useful*. The mean ratings show that participants found the engine overall useful for performing assessment. Table 4 shows the results ranked from highest to lowest with their mean rating and standard deviation.

Table 4: Usefulness - Ratings of EngAGe's web services

Feature	Rank	Mean	SD
Configuration file	1st	4.26	0.73
Player login	2nd	4.2	0.86
Score handling	3rd	4.11	0.82
Learning analytics	4th	4.09	0.84
Badges handling	4th	4.09	0.74
Feedback handling	6th	4.06	0.73
Leader board	7th	3.97	0.66
Game adaptation	8th	3.91	0.73

5.2.3 Usability

After confirming that EngAGe is useful to help define and perform a game's assessment, the evaluation focused on the usability of the tool. The evaluation questionnaire included three usability sections. First participants were asked, using five-point Likert scales, to rate the configuration file sections. Then, the same Likert scales were used to rate the usability of the web services available. Finally, an overall usability score was computed using the System Usability Scale (SUS) (Brooke, 1996) that has been proven to be robust and versatile (Bangor, Kortum, & Miller, 2008). Participants spent between 1 and 5.5 hours to complete the tutorial (average and median 3 hours) with between 30 minutes and 2 hours spent on writing the configuration file (average and median 1 hour) and between 45 minutes and 4 hours on the web services integration (average 2.4 and median 2.25 hours).

The configuration file

Each section of the configuration file was rated separately on a five-point Likert scale ranging from V*ery Easy* to V*ery Difficult*. Participants were asked the following question: "*Did you find the following sections of the configuration file easy to write and understand?*". The aims of this question were to determine whether the DSL was easy to use, the documentation was sufficiently detailed and whether some sections were too complex.

The mean ratings, calculated using scores ranging from 5 for *Very Easy* to 1 for V*ery Difficult*, show all sections highly rated. The *Evidence model* block containing the assessment logic was rated the lowest (3.51, SD = 0.94). This block contains all the logic for scoring and player actions and it requires more reflection on the part of developers. However, its mean is still above average. The *Game* block was the highest rated (4.23, SD = 0.9).

The web services

The usability of EngAGe's web services was evaluated using the same five-point Likert scale. Participants had to grade the eight main features of the engine. The answers took into account the integration of the web services calls to an educational game using EngAGe's library and the documentation associated with it (i.e. tutorial and online API). The results show that EngAGe is overall rated above average in terms of usability and no participant rated it as very difficult to use. Mean ratings range from 3.65 for the learning anaytics (SD = 0.87) to 3.86 for the badges handling (SD = 0.76).

System Usability Scale

The SUS questionnaire consists of 10 statements to be rated between "*Strongly Agree*" (5) and "*Strongly Disagree*" (1). Five statements are positive and five are negative and the questionnaire alternates between the two to avoid random answers. The scale includes a scoring system ranging from 0 to 100. Bangor et al. (2008) also propose a seven-point adjective rating scale for representing SUS scores ranging from "*Worst Imaginable*" to "*Best Imaginable*". The mean SUS score for the summative evaluation (n=36) is 64.38 with a median of 63.75 and a standard deviation of 13.86. According to the adjective rating scale, this value corresponds to a "*Good*" score. The details for each group of participants are shown in Table 5. As expected, and as shown in the next sub-section, experienced developers found EngAGe much easier to use than beginners.

Table 6 lists all the SUS statements and, for each one, the number of participants who elected *Agree* or S*trongly Agree*, the equivalent percentage, the mean ratings and standard deviation. The results are overall very positive; 27 participants (74%) thought that most educational games developers would learn how to use EngAGe quickly and 22 (61%) including all the experienced developers stated that they would like to use the system again. However, negative statements related to the learning curve and need for support are highly rated. 12 participants (33%) stated that they would need support to use it in the future and 4 (11%) that they had to learn a lot before using EnGAge.

Table 5: SUS score for each of the three groups of participants

Group	Mean	SD	Adjective rating
Experienced developers (n=7)	73.21	10.83	Excellent
Serious Game Honours students (n = 15)	68.17	11.6	Good
Game programming students (n = 14)	55.89	14.58	Good

Table 6: Participants' answers to the SUS questionnaire

Statement	Participants agreeing	Mean	SD
Positive statements			
I would imagine that most educational games developers would learn to use this system very quickly	27 (74%)	4	0.83
I found the various functions in this system were well integrated	23 (64%)	3.89	0.78
I think I would like to use this system in the future when developing educational games	22 (61%)	3.78	1
I thought the system was easy to use	17 (47%)	3.39	1
I felt very confident using the system	14 (39%)	3.11	1.2
Negative statements			
I think that I would need support to be able to use this system	12 (33%)	3.06	0.98
I needed to learn a lot of things before I could get going with this system	4 (11%)	2.69	1
I found the system unnecessarily complex	3 (8%)	2.42	0.92
I found the system very cumbersome to use	2 (6%)	2.39	0.87
I thought there was too much inconsistency in this system	0 (0%)	1.86	0.59

5.2.4 Comparing Results between the Participants

The overall results of the evaluation of usability and usefulness are positive. A Mann-Whitney *U* test was conducted to compare the opinions of participants based on their experience of programming. Three variables were analysed: the computed usability score, the mean usefulness of the configuration file's DSL and the mean usefulness of the web services. Experienced developers were grouped with Honours students and compared to novice game programmers. The results of the comparison are presented in Tables 7 and 8. They show that the means for usefulness and usability are higher in the experienced group. However, the Mann-Whitney *U* test only finds the difference to be highly significant for the usability score ($Z = -3.335$, $p = 0.01$). This suggests that, unsurprisingly, EngAGe is significantly easier to use for experienced developers but also that it is useful to all game developers.

Table 7: Descriptive statistics of developers' opinions on usability and usefulness based on their experience of programming

Experience		N	Minimum	Maximum	Mean	Std. Deviation
Novice	Usability Score	15	30.0	85.0	55.500	12.9284
	Usefulness of the DSL	15	2.00	5.00	3.6193	.93464
	Usefulness of the WS	15	2.43	5.00	3.8313	.77500

Experienced	Usability Score	21	45.0	95.0	70.714	11.2955
	Usefulness of the DSL	21	3.00	5.00	4.0648	.56160
	Usefulness of the WS	21	3.14	5.00	4.1533	.45361

Table 8: Result of Mann-Whitney *U* test for usefulness and usability of EngAGe based on developers' experience of programming

	Usability Score	Usefulness of the DSL	Usefulness of the WS
Mann-Whitney U	54.000	109.000	117.500
Wilcoxon W	174.000	229.000	237.500
Z	-3.335	-1.579	-1.290
Asymp. Sig. (2-tailed)	.001	.114	.197
Exact Sig. [2*(1-tailed Sig.)]	.001	.125	.202

6. Conclusions and future work

This paper has presented the research project EngAGe (an Engine for Assessment in Games), its background and motivations. The engine is used by developers when creating educational games resulting in a separation of the assessment from the game's mechanics. Educators can then, thanks to this modularity, modify the game's assessment and adapt it to their players via an online visual editor. For developers to integrate the engine in an educational game two main steps are required: (i) define the game's assessment in a configuration file using a domain-specific language (DSL); (ii) use a set of web services to perform the assessment in the game.

The generalisability, usefulness and usability of the engine have been evaluated with a pilot study and a summative evaluation involving 36 developers from various backgrounds. The pilot study introduced EngAGe in an existing educational game module at Higher Education level and allowed students to use the DSL and web services in the educational games they developed as part of the module. The summative evaluation consisted of a tutorial taking developers through the process of creating a flexible educational game, and a post questionnaire gathering information about their opinions of the creation of the configuration file and the use of the EngAGe web services.

Results concerning generalisability are promising as all 13 game development teams from the pilot study successfully defined their games using the proposed DSL. The usability evaluation returned a SUS score of 64, which corresponds to a "Good" score. Developers found both the DSL and the web services overall useful, however, the study has highlighted some areas that need improvement, such as the learning curve involved in using the engine. These findings suggest that EngAGe can be used efficiently and effectively by developers when creating educational games.

Another study was carried out with educators (Chaudy & Connolly, 2018) to evaluate the learning analytics dashboard of the engine and the visual editor that educators can use to modify a game's assessment. A tutorial was also be given to them and a post-questionnaire collected information about usability and usefulness. The educators' opinions towards educational games as an assessment tool was monitored to determine if the use of EngAGe influences it. Findings were overall very positive with educators rating the system useful and easy to use. Their trust toward using a game for assessment before and after using EngAGe also improved significantly.

Future work will include integrating the engine in various projects and at scale, working closely with game developers and collecting their opinion. The authors will also continue developing the engine, integrating the multi-player / peer-assessment feature, and offering external ways of logging players (e.g. using an LDAP server or Open Authorisation instead of a local authentication) to integrate with existing systems. Since May 2018, the General Data Protection Regulation (GDPR) is in effect in the European Economic Area (EEA), for the system to be used by external developers, tutorials on GDPR compliance will also be developed.

7. References

Annetta, L. A., Minogue, J., Holmes, S. Y., & Cheng, M.-T. (2009). Investigating the impact of video games on high school students' engagement and learning about genetics. *Computers & Education, 53*(1), 74-85.

Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction, 24*(6), 574-594.

Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry, 189*, 194.

Burgers, C., Eden, A., van Engelenburg, M. D., & Buningh, S. (2015). How feedback boosts motivation and play in a brain-training game. *Computers in Human Behavior, 48*, 94-103.

Buschang, R. E., Kerr, D. S., & Chung, G. K. (2012). Examining Feedback in an Instructional Video Game Using Process Data and Error Analysis. CRESST Report 817. *National Center for Research on Evaluation, Standards, and Student Testing (CRESST)*.

Carvalho, M. B., Bellotti, F., Berta, R., Curatelli, F., De Gloria, A., Gazzarata, G., . . . Martinengo, C. (2015). *The Journey: A Service-Based Adaptive Serious Game on Probability*, Cham.

Charsky, D., & Ressler, W. (2011). "Games are made for fun": Lessons on the effects of concept maps in the classroom use of computer games. *Computers & Education, 56*(3), 604-615.

Chaudy, Y., & Connolly, T. (2018). Specification and evaluation of an assessment engine for educational games: Empowering educators with an assessment editor and a learning analytics dashboard. *Entertainment Computing, 27*, 209-224.

Chaudy, Y., Connolly, T., & Hainey, T. (2013). *Specification and Design of a Generalized Assessment Engine for GBL Applications*. Paper presented at the European Conference on Games Based Learning, Porto.

Chen, S., & Michael, D. (2005). Proof of learning: Assessment in serious games. *Retrieved October, 17*, 2008.

Conati, C., & Manske, M. (2009). *Evaluating adaptive feedback in an educational computer game.* Paper presented at the Intelligent virtual agents.

Conlan, O., Hampson, C., Peirce, N., & Kickmeier-Rust, M. (2009). *Realtime Knowledge Space Skill Assessment for Personalized Digital Educational Games*. Paper presented at the Ninth IEEE International Conference on Advanced Learning Technologies, Riga, Latvia.

Csapó, B., Lörincz, A., & Molnár, G. (2012). Innovative Assessment Technologies in Educational Games Designed for Young Students *Assessment in Game-Based Learning* (pp. 235-254): Springer.

Filsecker, M., & Hickey, D. T. (2014). A multilevel analysis of the effects of external rewards on elementary students' motivation, engagement and learning in an educational game. *Computers & Education, 75*, 136-148.

Fowler, M. (2010). *Domain-specific languages*: Pearson Education.

Gander, S., & Parkway, R. (2000). Does learning occur through gaming. *Electronic Journal of Instructional Science and Technology, 3*(2), 28-43.

Garris, R., Ahlers, R., & Driskell, J. E. (2002). Games, motivation, and learning: A research and practice model. *Simulation & gaming, 33*(4), 441-467.

Girard, C., Ecalle, J., & Magnan, A. (2013). Serious games as new educational tools: how effective are they? A meta-analysis of recent studies. *Journal of Computer Assisted Learning, 29*(3), 207-219.

Göbel, S., Mehm, F., Radke, S., & Steinmetz, R. (2009). 80days: Adaptive digital storytelling for digital educational games. *2nd International Workshop on Story-Telling and Educational Games (STEG'09), 498*(498).

Hainey, T., & Connolly, T. (2013). *Development and Evaluation Of a Generic e-CLIL Web 2.0 Games Engine*. Paper presented at the 7th European Conference on Games Based Learning, Porto, Portugal.

Hainey, T., Connolly, T., Baxter, G., Boyle, L., & Beeby, R. (2012). *Assessment Integration in Games-Based Learning: A Preliminary Review of the Literature.* Paper presented at the 6th European Conference on Games Based Learning.

Harteveld, C., & Sutherland, S. C. (2015). *The Goal of Scoring: Exploring the Role of Game Performance in Educational Games.* Paper presented at the Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems.

Hoel, T., Griffiths, D., & Chen, W. (2017). *The influence of data protection and privacy frameworks on the design of learning analytics systems.* Paper presented at the Proceedings of the seventh international learning analytics & knowledge conference.

Jarvis, S., & de Freitas, S. (2009). *Evaluation of an immersive learning programme to support triage training.* Paper presented at the Games and Virtual Worlds for Serious Applications, 2009. VS-GAMES'09. Conference in.

Johnson, L., Adams, S., Cummins, M., Estrada, V., Freeman, A., & Ludgate, H. (2013). The NMC horizon report: 2013 higher education edition.

Kickmeier-Rust, M. D., Hockemeyer, C., Albert, D., & Augustin, T. (2008). *Micro adaptive, non-invasive knowledge assessment in educational games.* Paper presented at the Digital Games and Intelligent Toys Based Education, 2008 Second IEEE International Conference on.

Kickmeier-Rust, M. D., Mattheiss, E., Steiner, C., & Albert, D. (2011). A psycho-pedagogical framework for multi-adaptive educational games. *International Journal of Game-Based Learning (IJGBL), 1*(1), 45-58.

Killingsworth, S. S., Clark, D. B., & Adams, D. M. (2015). Self-explanation and explanatory feedback in games: Individual differences, gameplay, and learning. *International Journal of Education in Mathematics, Science and Technology, 3*(3), 162-186.

Lee, M. J., Ko, A. J., & Kwan, I. (2013). *In-game assessments increase novice programmers' engagement and level completion speed*. Paper presented at the Proceedings of the ninth

annual international ACM conference on International computing education research, San Diego, San California, USA.

Martinez-Ortiz, I., & Fernandez-Manjon, B. (2017). *Full Lifecycle Architecture for Serious Games: Integrating Game Learning Analytics and a Game Authoring Tool.* Paper presented at the Serious Games: Third Joint International Conference, JCSG 2017, Valencia, Spain, November 23-24, 2017, Proceedings.

McAlpine, M., van der Zanden, L., & Harris, V. (2010). *Using Games Based Technology in Formal Assessment of Learning.* Paper presented at the 4th European Conference on Games-Based Learning: ECGBL.

Metcalfe, J., Kornell, N., & Finn, B. (2009). Delayed versus immediate feedback in children's and adults' vocabulary learning. *Memory & Cognition, 37*(8), 1077-1087.

Orji, R., Vassileva, J., & Mandryk, R. L. (2013). LunchTime: a slow-casual game for long-term dietary behavior change. *Personal and Ubiquitous Computing, 17*(6), 1211-1221.

Perez-Colado, I. J., Perez-Colado, V. M., Martínez-Ortiz, I., Freire-Moran, M., & Fernández-Manjón, B. (2017). *UAdventure: The eAdventure reboot: Combining the experience of commercial gaming tools and tailored educational tools.* Paper presented at the Global Engineering Education Conference (EDUCON), 2017 IEEE.

Rick, S., & Weber, R. A. (2010). Meaningful learning and transfer of learning in games played repeatedly without feedback. *Games and Economic Behavior, 68*(2), 716-730.

Robbins, N. B., & Heiberger, R. M. (2011). *Plotting Likert and other rating scales.* Paper presented at the Proceedings of the 2011 Joint Statistical Meeting.

Rosas, R., Nussbaum, M., Cumsille, P., Marianov, V., Correa, n., Flores, P., . . . Salinas, M. (2003). Beyond Nintendo: design and assessment of educational video games for first and second grade students. *Comput. Educ., 40*(1), 71-94.

Sánchez-Mena, A., Martí-Parreño, J., & Aldás-Manzano, J. (2018). Teachers' intention to use educational video games: The moderating role of gender and age. *Innovations in Education and Teaching International*, 1-12.

Serge, S. R., Priest, H. A., Durlach, P. J., & Johnson, C. I. (2013). The effects of static and adaptive performance feedback in game-based training. *Computers in Human Behavior, 29*(3), 1150-1158.

Serrano-Laguna, Á., Martínez-Ortiz, I., Haag, J., Regan, D., Johnson, A., & Fernández-Manjón, B. (2017). Applying standards to systematize learning analytics in serious games. *Computer Standards & Interfaces, 50*, 116-123.

Shute, V. J. (2011). Stealth assessment in computer-based games to support learning. *Computer games and instruction, 55*(2), 503-524.

Siemens, G., & Gasevic, D. (2012). Guest Editorial-Learning and Knowledge Analytics. *Educational Technology & Society, 15*(3), 1-2.

Squire, K., Barnett, M., Grant, J. M., & Higginbotham, T. (2004). *Electromagnetism supercharged!: learning physics with digital simulation games*. Paper presented at the 6th international conference on Learning sciences, Santa Monica, California.

Thomas, J. M., & Young, R. M. (2010). Annie: Automated generation of adaptive learner guidance for fun serious games. *Learning*.

Thomas, P., Labat, J.-M., Muratet, M., & Yessad, A. (2012). *How to evaluate competencies in game-based learning systems automatically?* Paper presented at the Intelligent Tutoring Systems.

Torrente, J., Del Blanco, Á., Marchiori, E. J., Moreno-Ger, P., & Fernández-Manjón, B. (2010). < *e-Adventure>: Introducing educational games in the learning process.* Paper presented at the Education Engineering (EDUCON), 2010 IEEE.

Tsai, F.-H., Tsai, C.-C., & Lin, K.-Y. (2015). The evaluation of different gaming modes and feedback types on game-based formative assessment in an online learning environment. *Computers & Education, 81*, 259-269.

Vendlinski, T. P., Chung, G. K., Binning, K. R., & Buschang, R. E. (2011). Teaching Rational Number Addition Using Video Games: The Effects of Instructional Variation. CRESST Report 808. *National Center for Research on Evaluation, Standards, and Student Testing (CRESST)*.

Wiemeyer, J., Kickmeier-Rust, M., & Steiner, C. M. (2016). Performance Assessment in Serious Games. In R. Dörner, S. Göbel, W. Effelsberg, & J. Wiemeyer (Eds.), *Serious Games: Foundations, Concepts and Practice* (pp. 273-302). Cham: Springer International Publishing.

Wiersma, W. (2011). The validity of surveys: Online and offline: Oxford Internet Institute.

Zapata-Rivera, D., Hansen, E., Shute, V. J., Underwood, J. S., & Bauer, M. (2007). Evidence-based approach to interacting with open student models. *International Journal of Artificial Intelligence in Education, 17*(3), 273-303.